

Amdt. dated July 30, 2008  
Reply to Office action of August 4, 2006

Serial No. 10/629,939  
Docket No. SVL920020093US1  
Firm No. 0055.0064

**Amendments to the Drawings**

Submitted herewith is a replacement sheet for Sheet No. 9 to provide an amendment to FIG. 10 to change the reference number for the “current direction” element to 283 and to add the element “previous direction 282” to the cache 280.

Attachment: Replacement Sheet

## REMARKS/ARGUMENTS

Applicants provide a Replacement Sheet for Sheet No. 9 to amend FIG. 10 by changing the reference numeral for the “current direction” element from 282 to 283 and to add a “previous direction 282” element. Applicants submit that these amendments are disclosed in para. 38 of the Specification.

The claim amendments and arguments presented herein include claim amendments and arguments Applicants discussed with the Examiners during the phone interview on July 30, 2008, in addition to additional arguments. Applicants submit that the arguments and amendments presented herein make the substance of the phone interview of record to comply with 37 CFR 1.133. If the Examiner believes that further information on the interview needs to be made of record to comply with the requirements, Applicants request the Examiner to identify such further information.

In this Amendment, Applicants have amended claims and cancelled claim 2 and non-method claims 13-30 and 33-36 from further consideration in this application. Applicants are not conceding that the subject matter encompassed by claims prior to this Amendment is not patentable over the art cited by the Examiner. Claims were amended and cancelled in this Amendment solely to facilitate expeditious prosecution of the pending claims. Applicant respectfully reserves the right to pursue claims, including the subject matter encompassed by claims, as presented prior to this Amendment and additional claims in one or more continuing applications.

### 1. Claims 1-12, 31, and 32 are Patentable Over the Cited Art

The Examiner rejected claims 1-12, 31, and 32 as obvious (35 U.S.C. §103(a)) over Choy (U.S. Patent No. 5,551,027), Cornwell (U.S. Pub. No. 2002/0032678), and SGI (Linux Man page for fetch).<sup>1</sup> Applicants traverse with respect to the amended claims.

Amended claim 1 recites a method for accessing data in a database table, comprising: receiving a fetch request to fetch data from a base table that satisfies a query predicate, wherein rows of the base table are stored in table partitions and wherein there is one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node

---

<sup>1</sup> Although claims 31 and 32 were not listed as part of the obviousness rejection mentioned on page 3 of the Third Office Action, on page 10, the Examiner applied the art to reject claims 31 and 32 as obvious.

in each index partition includes at least one key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition; comparing a direction indicated in the fetch request and an ordering of the index partitions; setting a fetch direction based on a result of the comparison of the direction indicated in the fetch request and the ordering of the index partitions; scanning the index partitions in the fetch direction to determine a set of nodes from the index partitions whose key column value satisfies the query predicate; ordering the set of determined nodes from the index partitions; selecting one node from the ordered set based on a position of the node in the ordering; and returning data from the table row identified by the location identifier in the selected node in response to the fetch request.

Claim 1 is amended to add the requirements of comparing a direction indicated in the fetch request and an ordering of the index partitions and setting a fetch direction based on a result of the comparison of the direction indicated in the fetch request and the ordering of the index partitions; scanning the index partitions in the fetch direction to determine the index partitions. These added requirements are disclosed in at least paras. 39-41 and FIG. 11 of the Specification.

During the phone interview, the Examiner suggested Applicants amend the claims to clarify how the query is performed done to distinguish over the cited art. Applicants submit that these amendments focus on these concerns by specifying that certain operations be performed to determine the direction in which the partition indexes are scanned to determine nodes in the index partitions that satisfy the query.

The Examiner cited col. 11, lines 24-27 of Choy with respect to the claim requirement of determining a set of nodes, which as amended recites determining a set of nodes from a plurality of index partitions whose key column value satisfies the query predicate. (Third Office Action, pg. 3) Applicants traverse with respect to the amended claims.

Choy discusses a unique identifier of a partition, referred to as a “PID”. (Choy, col. 7, lines 19-21). Choy discusses a two-tiered indexing method that creates a Local Index Table for each partition of the database and creates a Coarse Global Index Table containing one unique global index entry for each distinct local index key value in each local index table. The local index table has one local index entry for each object of interest in the corresponding partition of

the table . (Choy, col. 8, lines 42-54) The global index table has the key values and for each key value the PID or local index having that key value. (Choy, col. 10, lines 25-31)

The cited col. 11 mentions that if there is a global index, qualified PIDs, i.e., PIDs of key values satisfying the query predicate, are obtained from that global index. The PIDs are sorted, duplicates are removed and the PIDs are merged with the PIDs based on the partition key. The query is then sent to each identified partition (PID). (Choy, col. 11, lines 37-40) The global index is used to direct the access request to the target partitions for processing. (Choy, col. 5, lines 19-42)

Nowhere do the cited col. 11 or other cited parts of Choy teach or suggest comparing a direction indicated in the fetch request and an ordering of the index partitions, setting a fetch direction based on a result of the comparison of the direction indicated in the fetch request and the ordering of the index partitions, and then scanning the index partitions in the fetch direction to determine index nodes satisfying the query. Instead, the cited Choy discusses processing a global index table to determine partitions (PIDs) having key values satisfying the query and then sending the queries to the local indexes to handle. For instance, the Examiner has not cited any part of Choy that teaches comparing the direction indicated in the fetch or query request and the ordering of the local indexes of Choy to determine a direction in which the indexes are scanned for matching entries.

The Examiner cited col. 1, lines 45-55 of Choy as teaching the claim requirement of ordering the set of nodes determined from multiple index partitions. (Third Office Action, pg. 3) The cited col. 1 mentions that the nodes may include pointers to records in the database, where the pointers include additional key record information that may reference the records. The record keys are stored in an ordered form through the nodes at branches in the tree. Although the cited col. 1 mentions that nodes may be ordered, the cited Choy does not teach or suggest determining nodes from the index partitions whose key column value satisfies the query predicate and then ordering the determined nodes determined from the index partitions. For instance, the Examiner has not cited any part of Choy that teaches ordering the results of the Local Indexes.

The Examiner cited col. 11, lines 43-47 of Choy as teaching the claim requirement of selecting one node from the ordered set based on a position of the node in the ordering. (Third Office Action, p. 3). The cited col. 11 discusses how each node having a local index may use a

different query evaluation plan to evaluate the query. Nowhere does this teach or suggest selecting one node from an ordered set, where the ordered set comprises a set of nodes comprising entries in index partitions that satisfy the query predicate. Moreover, the cited “nodes” of Choy are different than the claimed nodes because the cited nodes of Choy comprise a computational node having a local index (see, Choy, col. 4, line 66 to col. 5, line 5, col. 5, line 58-62) on which a query is performed, whereas the claimed nodes comprise nodes in from different index partitions that have a key column value identifying a corresponding table row in the corresponding table partition.

The Examiner cited SGI and Cornwell as teaching fetch request processing and found that it would be obvious to modify Choy’s global index and local index processing scheme to be applied to fetch request processing. (Third Office Action, pgs. 3-4) Applicants submit that even if one made this modification, this proposed modification still fails to teach or suggest the above discussed deficiencies of Choy in that the claimed technique for comparing the direction indicated in the fetch request and the ordering of the index partitions and setting a fetch direction to scan the indexes based on the result of the comparison is different from the cited scheme of Choy of using a global index and local indexes. Instead, the cited SGI and Cornwell discuss fetch requests, but do not teach or suggest processing a fetch request by comparing the direction of the fetch request with the ordering of index partitions as claimed.

Moreover, Applicants submit that there is still no teaching of using the claimed index partitions to determine qualifying nodes for a fetch request as claimed. The Examiner cited col. 2, lines 55-59 of Choy as providing the motivation to modify the index partitioning scheme of Choy to be applied to processing fetch requests. (Third Office Action, pg. 4) The cited col. 2 mentions that indexes are maintained on a search field to provide search efficiency. Applicants submit that although indexes provide search efficiency, there is no teaching or suggestion here that the particular described index scheme of Choy be used for fetch requests or perform the searching of the index partitions as claimed. For instance, there is no suggestion or motivation to use with search requests the claimed indexing scheme of one index partition for each determined table partition, comparing the direction indicated in the fetch request with the ordering the index partitions.

With respect to canceled claim 2, which recited modifying the direction in which the index partitions are scanned, the Examiner cited he Examiner cited paras. 93, 99, and 100 of

Cromwell. (Third Office Action, pgs. 4 and 14) Applicants submit that these cited paras. 93, 99, and 100 do not teach or suggest the requirements added to claim 1 of comparing the direction indicated in the fetch request and ordering of the index and setting the fetch direction in which the indexes are scanned based on the result of the comparison.

The cited paras. [0093] and [0099] discuss that cursors can fetch forward and para. [0094] mentions that the database program may perform a FETCH backwards operations to fetch backwards. The cited para. 100 discusses converting a FETCH absolute into a FETCH relative having a k equal to the determined relative distance in the fetch absolute.

Although the cited paragraphs of Cromwell discuss fetching in a forward and backward direction, nowhere do the cited paragraphs anywhere teach or suggest comparing a direction indicated in the fetch request with an ordering of index partitions to set a fetch direction, which is the direction in which the index partitions are scanned to determine nodes that satisfy the query. Instead, the cited paragraphs discuss fetching in a direction, but not setting the fetch direction in which index partitions are scanned based on the comparison of the direction of the fetch request and the ordering of the index partitions.

Accordingly, claim 1 is patentable over the cited art because the cited references do not teach or suggest the combination of claim requirements.

Claims 3-12, 31, and 32 are patentable over the cited art because they depend from claim 1, which is patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional ground of patentability over the cited art.

Amended claim 3 depends from claim 1 and further recites that the fetch direction is set opposite the direction indicated in the fetch request if the direction indicated in the fetch request is opposite the ordering of the index partitions.

Claim 3 was amended to depend from claim 1 and to specify that the fetch direction is set opposite the direction in the fetch request if the direction in the fetch request is opposite the index partition ordering. The added requirements of claim 3 are disclosed in at least paras. 39-41 and FIG. 11 of the Specification.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of pre-amended claim 3. (Third Office Action, pg. 4). Applicants traverse.

The cited para. [0099] mentions performing a FETCH ABSOLUTE of k, where k is the number of rows to fetch forward for a +k or negative (-k). The data manager determines the

absolute row number of the entry in the result table pointed to by the cursor and the page including the entry, and determines the relative distance of the requested entry from the current entry as the absolute value of K.

Although the cited para. [0099] discusses how to determine the relative distance of a current result table entry to the entry to which the fetch command wants to fetch, the cited paragraph nowhere teaches or suggests setting the fetch direction in which the index partitions are scanned opposite the direction indicated in the fetch request if the ordering of the index partitions is opposite the direction in the fetch request. Instead, the cited para. [0099] discusses determining a relative distance to fetch. There is no teaching or suggestion of determining whether to modify the fetch direction in which the index partitions are scanned based on whether a current fetch direction is opposite an ordering of the index partitions.

Accordingly, claim 3 provides additional grounds of patentability over the cited art because the additional requirements of claim 3 are not taught or suggested in the cited art.

Amended claim 4 recites that setting the fetch direction comprises: setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions; and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions.

Applicants amended claim 4 to depend from claim 1 and to clarify that the operation of setting the fetch direction comprises the claimed setting operations to conform claim 4 with amended claim 1.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of claim 4. (Third Office Action, pgs. 4-5) Applicants traverse for the following reasons.

The cited para. [0099] discusses a fetch absolute operation, which is the number of rows to fetch forward or backward from the first entry in the result table. This is performed by determining the distance from the current entry to the requested entry, and then convert this command to a fetch relative to fetch to the requested position.

Nowhere does the cited para. [0099] anywhere teach or setting the fetch direction in which the index is scanned based on the partition index ordering and the direction indicated in

the fetch request. Instead, the cited para. [0099] discusses how to fetch to an absolute requested row from the current position, not to change the direction of a fetch based on the index ordering.

Further, the Examiner has not cited any part of Cromwell and paragraphs of setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions. These specific operations to set the direction based on the fetch direction and the ordering of the index partitions is nowhere taught or suggested in the cited art.

Accordingly, claim 4 provides additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Amended claim 5 depends from claim 1 and further requires that if the fetch request is a first fetch of the fetch request, then selecting one node starting from one of: a lowest key value from each index partition if the fetch direction is forward or highest key value from each index partition if the fetch direction is backward.

Claim 5 was amended to depend from claim 1.

The Examiner cited para. [0087] as teaching the additional requirements of these claims.  
(Third Office Action, pg. 5)

The cited para. [0087] discusses positioning the cursor to the position specified in the fetch, e.g., prior, first, last, current, etc if the fetch is insensitive and then returning the row positioned at the cursor. If the returned row was previously fetched, with a fetch sensitive it would reflect any changes made to the base table prior to the fetch sensitive operation.

Although the cited para. [0087] discusses how to position the cursor on a row depending on the fetch specified, there is no teaching or suggestion of selecting a node from a lowest or highest key value from each index partition depending on the fetch direction. There is no mention in the cited para. [0087] of selecting key values from index partitions as claimed. Instead, the cited para. [0087] discusses how to fetch to a requested position and then return the data from the row to which the cursor points, which may comprise changed data if the previous fetch was fetch sensitive.

Accordingly, claim 5 provides additional grounds of patentability over the cited art because the additional requirements of claim 5 are not taught or suggested in the cited art.

Amended claim 6 depends from claim 1 and further recites if the fetch request is not a first fetch of the fetch request, then determining whether the fetch direction in which the index partitions are scanned for a previous fetch request is a same direction as the direction indicated in a current fetch request, wherein the direction indicated in the fetch request is capable of having been modified; and if the fetch direction for the previous fetch request and direction indicated in the current fetch request are different, then discarding all saved nodes for the index partitions and selecting one node from a last selected node.

Claim 6 was amended to depend from claim 1 and to clarify the direction indicated in a previous and current fetch request and the fetch direction in which the index partitions are scanned for the previous fetch request. These added requirements are disclosed in at least paras. 39-41 and FIG. 11 of the Specification.

Claims 6-10 provide further limitations concerning the direction of the fetch request and index partitions. The Examiner cited other sections of Cromwell as teaching the additional requirements of these claims that concern how to fetch in different directions in a result table to retrieve a row from a result table. Nowhere does the cited Cromwell teach or suggest the additional requirements of these claims providing additional requirements concerning index partitions on table partitions. Instead, the cited Cromwell discusses how to fetch forward or backward through a result table whose sequential rows are on multiple pages.

Claim 11 depends from claim 1 and further requires discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request; determining a new set of nodes from each index partition; and caching the determined new set of nodes when performing the fetch operation.

The Examiner cited para. [0097] of Cromwell as teaching discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request. (Third Office Action, pg. 7) Applicants traverse.

The cited para. [0097] discusses how the data manager fetches a number of pages from storage. The database program uses a statistical consideration to determine whether rows are being sequentially accessed in order to determine whether to prefetch improve performance. Although the cited para. [0097] discusses using a statistical algorithm to determine whether to

prefetch for sequential access, the Examiner has not cited any part of para. [0097] that teaches or suggests discarding cached keys if the fetch request is in an opposite direction of a previous fetch request. The cited para. [0097] discusses detecting sequential access. However, there is no mention or teaching of the claim requirement of discarding cached keys if the fetch request is in an opposite direction of the previous fetch request.

Accordingly, claim 11 provides additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1, 3-12, 31, and 32 are patentable over the art of record. Should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: July 30, 2008

By: /David Victor/

David W. Victor  
Registration No. 39,867

Please direct all correspondences to:

David Victor  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: 310-553-7977  
Fax: 310-556-7984